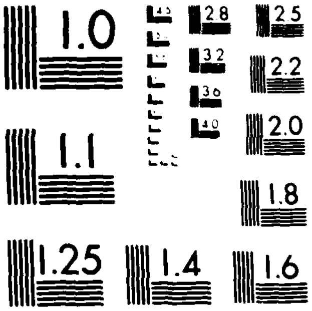
AD-A121 182   RESEARCH IN VLSI SYSTEMS HEURISTIC PROGRAMMING PROJECT   1/1
AND VLSI THEORY PR..(U) STANFORD UNIV CA INTEGRATED
CIRCUITS LAB J HENNESSY ET AL. NOV 82 MDA903-79-C-0680

UNCLASSIFIED   F/G 9/2   NL

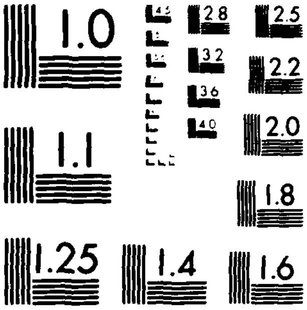
END

FILMED

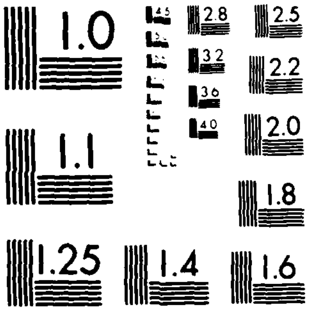
DTIC



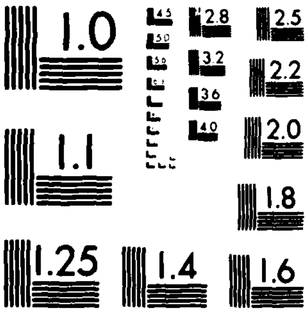
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A



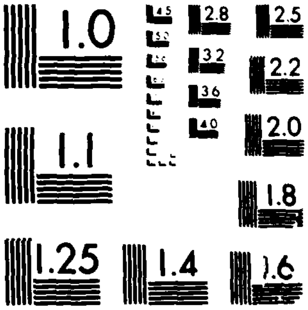
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

Computer Science Department
Computer Systems Laboratory
Information Systems Laboratory
Integrated Circuits Laboratory

STANFORD UNIVERSITY · STANFORD, CA 94305

TECHNICAL STATUS REPORT
1 May 1982 to 31 October 1982

Research in VLSI Systems
DARPA Contract No. MDA903-79-C-0680
DARPA Order No. 3773
Principal Investigators: J. Hennessy, T. Kailath
Sponsored by Defense Advanced Research Projects Agency
Monitored by P. Losleben

Heuristic Programming Project and VLSI Theory Project
DARPA Contract No. MDA903-80-C-0107
DARPA Order No. 3423
Principal Investigators: E. A. Feigenbaum, B. G. Buchanan, J. D. Ullman
Sponsored by Defense Advanced Research Projects Agency
Monitored by R. Ohlander

A Fast Turn Around Facility for Very Large Scale Integration (VLSI)
DARPA Contract No. MDA903-80-C-0432
DARPA Order No. 3709
Principal Investigator J. T. Meindl
Monitored by D. Adams

"The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the offical policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government."

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER MDA903-79-C-0680(2)	2. GOVT ACCESSION NO. AD-A121 182	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Research in VLSI Systems Heuristic Programming Project & VLSI Theory Project A Fast Turn Around Facility for Very Large Scale Integration (VLSI)		5. TYPE OF REPORT & PERIOD COVERED Technical Status Report May 1982-October 1982
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) J. Hennessy, R. Matthews, J. Newkirk, J. Shott, J. Ullman, H. Brown		8. CONTRACT OR GRANT NUMBER(s) MDA903-79-C-0680 MDA903-80-C-0107 MDA903-80-C-0432
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Science Dept.; Computer Systems, Information Systems, Integrated Circuits Laboratories; Stanford University, Stanford, CA.		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency Arlington, Virginia 22209		12. REPORT DATE November 1982
		13. NUMBER OF PAGES 31
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research (Stanford Branch) Stanford University Stanford, California 94305		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Unclassified; approved for general distribution.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Very Large Scale Integration	Computer Supported FTL
Regular Expression Compilation	Palladio: IC Designer's Assistant
MIPS: A VLSI Processor	Electron Bean Lithography
Relative Layout Tools	Micron CMOS
Graphics Architectures	Wafer FabricationFacility

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Table of Contents

A

Executive Summary

The major progress of note for this period is as follows:

1. *Regular Expression Compilation.* The overall goal of the project is to develop silicon compilers that produce output comparable to hand designs. A compiler for translating a mixture of state machine definition and regular expressions into networks of PLA's or logic is working. In a number of tests the area required by the output was found to be no more than 50% over that of a hand design; in some cases the results are far closer than that.

2. *MIPS: A VLSI Processor.* MIPS (Microprocessor without Interlock between Pipe Stages) is a project to develop a high speed (> 1 MIP) single chip 32-bit microprocessor. The final test chips for the MIPS processor design were completed and will be submitted for fabrication on the November 82 MOSIS fabrication run.

3. *Relative Layout Tools.* Yale (Yet Another Layout Editor) is a symbolic layout editor that will run on the SUN and make the capabilities of SILT available in a graphics front-end. The first version of Yale was completed and documented.

4. *Graphics Architectures.* The IRIS is a high-resolution, high-performance, color-graphics workstation. It incorporates the Geometry Engine and utilizes the SUN processoir board. An IRIS prootype was designed and demonstration software was developed.

5. *Computer supported FTL launched.* We have completed the planning and exploratory stages of a project to provide extensive automation and computer support for the Fast Turnaround Laboratory. This ambitious interdisciplinary project (involving researchers from Computer Systems, Integrated Circuits, and Solid State laboratories) will provide control, documentation, training, portability, repeatability, and efficiency in the area of IC fabrication processes.

6. *Palladio: An IC Designer's Assistant.* The Palladio system is a framework for experimentation with circuit design methodologies, knowledge-based expert system design aids and symbolic circuit simulation concepts. The major goal of the project is to develop an intelligent and integrated circuit design environment to assist in the full design, test and debug design cycle. During the past six months Palladio's basic system structure was completed and several prototypic system components were implemented.

7. *Electron Beam Lithography.* The Stanford MEBES machine has passed all on-site acceptance tests, including those relating to direct writing and registration accuracy. The MEBES machine is currently being used to fabricate masks for the Geometry Engine, several MIPS test chips, and the two-chip Medium Tester.

8. *2 Micron CMOS.* Test devices using Stanford's 2 μm CMOS process have been fabricated. This process features a 4 μm n-well and a 400 Angstrom oxide thickness. Two additional "tune-up" runs are in progress at the moment. The mask set for this run was written at Perkin-Elmer/ETEC before our machine passed acceptance, but plates were developed and etched at Stanford.

9. *Wafer Fabrication Facility.* The following pieces of equipment have been installed and characterized and are now being used in the fabrication of NMOS/CMOS wafers: Drytek 100 plasma etcher (Poly-Si and Si_3N_4); Tylan LPCVD (Poly-Si, Si_3N_4, and LT SiO_2); MTI Omni-Chuck resist processing; IPC Σ-80 plasma etcher (SiO_2); and Ultratech 900 1:1 projection aligner.

10. *Tri-Level Resist Technology.* A tri-level resist technology designed for direct-write E-beam lithography has been developed. The three layers are: 1.2 μm "super-hard-baked" AZ-1470 resist, 500 Angstrom sputtered or plasma-deposited poly-Si, and 0.4 μm PBS electron resist. The key step in this process is the use of O_2 reactive ion etching (RIE) to transfer to pattern from the thin PBS and poly-Si layers to the underlying "planarization" layer of AZ-1470 resist. Resist lines 0.5 μm wide separated by 0.5 μm have been produced by the highly anisotropic RIE.

11. *Testing for Process Control.* A number of test structures have been designed which allow statistical data to be gathered about many of the parameters affecting fabrication yield including step coverage, shorts (both level-to-level and on a single level), contact hole integrity, etc. In contrast to many of the "string" or "meander" structures, these test vehicles are addressable in such a way as to allow the position information of failures to be monitored.

12. *Electrically based layout system, Lava.* A rewritten version of Lava is again running test cases, including the 10,000-transistor serial memory. It seems stable enough to support further investigations, *e.g.*, composition of cells and logic-to-sticks conversion.

13. *Logic-to-sticks conversion, Dumbo.* Dumbo produced its first totally automatic layouts with reasonable area efficiency, using force-directed placement. Large cells still incur large area penalties, however.

14. *The MEDIUM tester chip set.* The MEDIUM tester chip set has been designed, laid out, and submitted for fabrication, along with some test chips. One of the two main chips has been partially tested, and it appears to be correct.

15. *Bulk CMOS.* We have implemented a CMOS design-rule checker based on the polygon package and distributed it to MIT and JPL. It was used to check pads, PLAs, and a counter submitted for fabrication; it has also checked layouts from MIT and Lincoln Labs.

Technical Progress

1 Design Description, Analysis, and Synthesis

1.1 Regular Expression Compilation

A system compiling regular expressions into PLA's or logic has been developed. The input language has been augmented recently to include state declarations when convenient; in the syntax, entering a state looks similar to the occurrence of an input symbol, while transfer to a state is akin to emitting an output symbol.

The regular expression language is translated to a nondeterministic finite automaton (NFA) language by one of two different compiler strategies, called "before" and "after." The former tends to minimize the number of rows of a PLA, while the latter tends to minimize the columns. Neither strategy dominates the other in tests, so both are made available as options for the user.

The NFA's are compiled into networks of PLA's or Weinberger arrays (via S. Johnson's lgen language). The PLA's are optimized using GRY [Hemachandra 82]. Layout of PLA's is accomplished by PLAGEN, a routine written in CHISEL [Karplus 82]. The latter two facilities are the product of K. Karplus, a Hertz fellow whose DARPA support consisted of computer services.

Staff: L. Hemachandra, A. Karlin, H. Trickey, J. Ullman

Related Efforts: lgen (Bell Labs), SLIM (Stanford)

References: [Hemachandra 82, Floyd 82, Trickey 82, TrickeyUllman 82, Ullman 82a]

1.2 YALE

Yale (Yet Another Layout Editor) is a symbolic layout editor that will run on the SUN and make the capabilities of SILT available in a graphics front-end. YALE is presently being implemented on a combination of the SUN workstation and the VAX. It uses the SUN as an intelligent graphics workstation (no disc required); thus, this work is being carried out in collaboration with the Network Graphics project at Stanford. YALE is primarily a graphics interface to SILT, allowing the placement of *reference lines* graphically. It also allows textual or graphical specification of *constraints* and textual specification of expressions for computation of reference line placement.

Staff: J. Clark, T. Davis

Related Efforts: Daedalus and the Data Path Generator (MIT), Caesar (UCB).

References: [Davis, T. and Clark, J. 82]

1.3 SLIM

SLIM, Stanford language for Implementing Microcode, was initially implemented during an earlier contract and presented at the 1981 Caltech VLSI Conference. The goals of SLIM are to describe on-chip control as microcode, to simulate that microcode using a functional description of the chip components, and to generate a PLA implementation of the microcode. The initial SLIM implementation has been working since the end of 1980.

The current work on SLIM concentrates on the design of a state-assignment optimizer. A prototype optimizer, which saves an average of 15% of the minterms, has been developed. It needs further work to characterize its theortical properties and to make it more efficient on large PLA's.

Staff: J. Hennessy, L. Adams

Related Efforts: MacPitts (Lincoln Labs), SLANG (UCB) and SLAP (Brown University).

1.4 Palladio: An IC Designer's Assistant

The Palladio system [Brown 82] is a framework for experimentation with circuit design methodologies, knowledge-based expert system design aids and symbolic circuit simulation concepts. The major goal of the project is to develop an intelligent and integrated circuit design environment to assist in the full design, test and debug cycle. Palladio serves as the focus for the Knowledge-based VLSI Project (KBVLSI project), a collaborative activity between the Heuristic Programming Project, Stanford University, the VLSI System Design Area, Xerox Palo Alto Research Center and the Fairchild Laboratory for Artificial Intelligence Research.

1.4.1 Design Specification Editor

A circuit design process can be viewed as the creation of behavioral and structural specifications of the circuit. This usually involves a sequence of transformations from abstract specifications of the behavior and structure to more detailed implementation specifications. Palladio's design editor is an interactive graphics system for creating and editing circuit specifications at various levels of structural and behavioral detail. For example, in Palladio the structure of a circuit component can be simultaneously described in terms of gates, in terms of switches and in terms of a layout, and the component's behavior can be simultaneously described in terms of boolean logic or in terms of node value-strength pairs.

The level of specification to be used in performing a particular task is currently at the choice of the designer. We are working on design aids which will automatically select the most appropriate specification levels for the task at hand, for example, simulation.

During the past six months we have essentially completed the the underlying framework for Palladio's design editor and the graphical user interface to the editor. The graphical interface uses HILGA [Gerring 81], a high level graphics package for Interlisp-D.

Staff: H. Brown, G. Foyster, P. Gerring.

1.4.2 Design Specification Levels

One of the objectives of our project is to experiment with various circuit design specification levels. In Palladio a design specification level is primarily represented by structure and/or behavior specification languages (both graphical and textual) and by a set of composition rules for governing the recursive creation of composite components from the primitives of the specification language [Stefik 82a]. During the past six months we have implemented two specification levels.

The Clocked Primitive Switches (CPS) level is a structural specification level which includes both circuit and gate level specifications. In addition the graphical form of the CPS level can be used to describe a planar topology for the circuit.

Associated with the the CPS structural level are two behavioral specification levels. One is based on the usual boolean level of behavioral specification and the other is based on node value-strength pairs [Bryant 81]. The user interface form of both of these behavioral levels is production rules.

Staff: H. Brown, G. Foyster.

The Computational Blocks Abstraction (CBA) specification level is a level at which a designer can specify a digital system in terms of blocks containing data structures and operations.

Staff: C. Tong.

1.4.3 Design Simulator

During the past six months we have been working on a framework for multiple-level, rule-based simulator. The simulator is not tied to any particular technology or specification level. The primary idea is to exploit hierarchical design descriptions to help manage the simulation of complex systems. The simulator framework can be used to perform auto-validation of designs, goal-directed simulation, and symbolic simulation.

A preliminary version of the simulator framework has been implemented. This implementation was done in MACLISP using the Meta-level Representation System (MRS) [Genesereth 80]. MRS provides, in partiular, the simulator's inferencce mechanisms.

Staff: N. Singh (Stanford and Fairchild).

The simulator framework has been interfaced with the CPS specification level in Palladio, and a dynamic, graphical simulator display has been implemented.

Staff: G. Foyster.

1.4.4 Design Transformation

During the past year we have been working on design aids to assist in the transformation of an abstract design specification to a more detailed implementation. This work has involved research on a model for design centered around goals, alternative designs and tradeoffs [Tong 82]. This model views design as a dialectic between goals and design alternatives: goals are established, alternative designs are specified, and the goals are evaluated with respect to these alternatives and possibly revised in light of the alternatives. Knowledge of important tradeoffs among goals are used to guide the dialectic.

To support the complex and varying relationships among design entities a design knowledge representation language, FIRE, has been developed. FIRE is implemented in LOOPS [Bobrow 81].

Staff: C. Tong.

1.4.5 Programming Systems

Most of the Palladio system is implemented in LOOPS. LOOPS is a data and object oriented programming system integrated with Interlisp. In object oriented programming, behavior is determined by responses of instances of classes to messages sent between these objects with no direct access to the internal structure of an object. Data oriented programming is a dual of object oriented programming where behavior can occur as a side effect of direct access to object state.

During the past six months numerous enhancemens and extensions have been made to LOOPS. In particular, the capability for doing rule oriented programming has been added to LOOPS [Stefik 82b].

Staff: D. Bobrow (Xerox), M. Stefik (Xerox).

Some of the project programming has utilized the GLISP compiler system [Novak 82]. During the past six months GLISP has been extended and made more robust, and a graphics editor based on GLISP object descriptions has been developed for the Xerox D1100 Workstation (the Dolphin).

The GLISP language has been extended as an experimental hardware description language. This language allows hardware data structures such as computer instruction formats to be described and used in a natural way. The GLISP descriptions are compiled into an intermediate code which is similar to exisiting register tranfer languages. This intermediate code runs on an interactive simulator using the Dolphin graphics system.

Staff: G. Novak.

1.5 Electrically based layout system, Lava

Lava is an electrically based, general-purpose layout language. Our principal objectives are topological, rather than geometric, layout description and guaranteed design-rule correctness of layouts. Lava's major components are a sticks compactor, cell stretching and abutment mechanisms, a router, and a framework to link them together.

We have rewritten Lava to stabilize it and to incorporate some of the hooks that will be necessary for further investigations. We have concentrated on a clean implementation of the aspects that we understand well, removing some of the more ill-conceived mechanisms in the previous implementation. One major improvement is that much of the technology-specific information is now centralized; while Lava is not intended to be technology-independent, this technology file makes it possible to change easily parameters of the nMOS target process.

The result is a sufficiently stable platform for further investigations, for example, a well-conceived composition level and logic-to-sticks conversion. The rewritten Lava now successfully compiles a large number of test cases, including (very recently) the serial memory chip described in our previous report.

Staff: C. Burns, D. Chapiro, P. Eichenberger, R. Mathews, J. Newkirk, D. Perkins, T. Saxe

Related Efforts: EARL (CalTech), CABBAGE (UCB)

1.6 Routing

We have developed a new, 2-dimensional area router, the loop routing scheme (LRS). LRS handles both rectangular- and doughnut- shaped routing areas. LRS is a promising box router for the custom routing problem because, like the dogleg channel router, it indicates how much expansion of the routing area is necessary to complete a given routing.

The difficulty of channel routing problems, and the performance of channel routers, may be measured by the number of wiring tracks required to complete the routing. Previously, no similar measures existed for comparing area routers. Such a measure must describe how difficult a given, fixed, area-routing problem is, since there is no well-defined way to expand the routing area to guarantee completion of the route.

We have developed an appropriate measure of problem difficulty, the Manhattan Area Measure. By using it to assess the difficulty of routing problems generated using Monte Carlo techniques, we have compared the performance of LRS to the classic Lee area-routing algorithm. The LRS has vastly superior performance to the Lee, successfully routing problems that are twice as dense as those that the Lee will complete successfully.

Staff: T. Saxe, L. Smith

Related Efforts: PI project (MIT)

References: [Smith 82]

1.7 Logic-to-sticks conversion, Dumbo

This work is aimed at simplifying the layout of random logic. Some amount of glue is inevitable in a design, but is is painful to lay out and typically does not consume a significant amount of the total area of the design. Consequently, we are searching for techniques for converting logic, specified as transistors, gates, and a net list, to stick diagrams for compaction by Lava.

Th·· logic-to-sticks conversion program Dumbo has now produced some layouts of small cells with respectably small areas completely automatically. For example, a 12-transistor cell was laid out with no area penalty when compared to the original hand-drawn sticks. Force-directed placement and orientation of components made this result possible. However, for larger cells we still see penalties of 100-150%.

We are continuing to analyze the sources of inefficiency in Dumbo layouts. As for our custom-chip router described in our previous report, a series of small 10% efficiencies cumulate to create a large overall area penalty. We are analyzing these sources of error to try to understand and control them. However, one major source of inefficiency appears to be the sensitivity of our sticks compactor to small changes in the stick diagram. While the human designer copes with these vagaries very well, we do not understand them well enough to permit us to avoid them in automatically generated cells. Nevertheless, we feel we will be able to make substantial additional progress in this area.

Staff: R. Mathews, D. Perkins, W. Wolf

Related Efforts: Rule-based circuits-to-sticks conversion (A. Bell, PARC)

References: [Wolf 82]

2 VLSI Processor Architecture

2.1 MIPS - A High-Speed Single-Chip VLSI Processor

MIPS (Microprocessor without Interlock between Pipe Stages) is a project to develop a high speed (> 1 MIP) single-chip 32-bit microprocessor. Like the RISC project at Berkeley, MIPS uses a simplified instruction set and is a load-store architecture.

The MIPS architecture is summarized in a previous technical progress report and is discussed in several publications. During this six month period, the major goal of the project was to develop a series of test chips that would provide a complete test of each major component of the chip individually.

The six test chips contain all the parts needed to implement the complete MIPS processor. Each test chip also contains additional testing and pin multiplexing hardware. By fully testing the components before fabricating a complete design, the probability of success on the first run is much higher. This approach also allows us to characterize the indivdual components and make performance adjustments before the final fabrication. By designing a single reusable test frame, the individual test chips may be constructed from the exact pieces of the complete chip with a minimal amount of effort. The final assembly process consists of merely composing the individual test components to form the complete processor. Lastly, this process offers an ideal opportunity to test the concept of fast-turnaround foundries. Because progress on the project depends on receiving and testing the chips prior to completing the final design, reasonable quality, fast-turnaround fabriction is essential.

The six test chips and their current status is as follows:

1. Register File Test Chip - submitted and tested at both 3 and 4 microns. The 4 micron chips were functional, although the yield was <10% (i.e. all parts of the chip worked over 10 die, but no single die was completely functional). The 3υ fabrication produced no working parts.

2. Instruction Decode Test Chip - tested one out of eight die was completely functional.

3. Barrel Shifter Test Chip - preliminary tests from the first fabrication have shown partially working chips. So far, no definitive problems have been identified.

4. ALU Test Chip - destined for November submission. Currently running functional simulation.

5. Program Counter Test Chip - version 0 currently in test. Version 1 in simulation.

6. Master Pipeline Control Test Chip - submitted in October fabrication run. Not returned yet.

Staff: F. Baskett, J. Burnett, J. Gill, K. Gopinath, T. Gross, J. Hennessy, N. Jouppi, W. Park, S. Przybylski, C. Rowen, A. Strong.

Related Efforts: RISC (UCB), IBM 801 (IBM Yorktown), Cray-II (Cray Research).

References: [HennessyJouppiPrzybylski 82, Gross 82, Hennessy 83]

2.2 Geometry Engine

The Geometry Engine is a high-performance, floating-point computing engine for geometric operations in 2D and 3D computer graphics. Multiple copies of the Geometry Engine provide a parallel computing system with very high-performance. (5-10 million floating-point operations per second.)

During this period, we obtained a second fabrication of Geometry Engines. This batch provided enough chips to build a geometry system (10 chips) and a complete prototype. This prototype system, called the IRIS, is discussed in the next section. The Geometry Engine design is completely functional, although the performance is less than originally predicted.

Staff: J. Clark, M. Hannah

References: [Clark 82]

2.3 IRIS Workstation

The goal of the IRIS workstation project was to design a high-resolution, color, extra high-performance graphics workstation that utilized all of the features of the Geometry Engine and was software-compatible with the SUN 68000 processor (excluding graphics software compatibility). The system consists of

- A SUN-compatible processor/memory board.
- A Geometry Engine board (10 Geometry Engines).
- A Raster Generation Subsystem.
- A Raster Update Subsystem.

The IRIS allows the user program to generate display programs that provide for real-time motion of 2D and 3D environments, multi-window displays and color lookup table manipulation. To provide for motion simulation, the system is dynamically configurable to provide either double or single-buffer images. The system has been in operation since August, 1982, and procedures are underway to replicate copies of the system for future research at various Stanf rd Laboratories.

Staff: K. Akeley, J. Clark, M. Grossman, C. Rhodes

3 Signal Processing Algorithms and Architectures

3.1 Simulation of Musical Instruments

The Digitar chip uses a digital-filter method to synthesize various waveforms. It consists of about 3500 transistors, and was designed by writing a SAIL program to generate its layout. Internally, it has a 12-bit arithmetic unit that adds, subtracts, increments, decrements, and single-bit shifts; it has a 16 by 15-bit shift register array and a 7 by 21-bit microcode array for controlling the data path. Although primarily designed to synthesize plucked string sounds, the chip can also produce snare-drum, clarinet, and bassoon timbres. It is controlled by an 8-bit port and internally decodes the command sequences sent to it by a microprocessor. The chip requires 4K of external RAM in order to perform its functions, and it addresses this RAM directly. *Support for this project was primarily by the Hertz Foundation; ARPA support has been limited to* computation facilities and a small amount of personnel support.

Staff: K. Karplus, A. Strong

4 Testing

4.1 Process Control Test Structures

Work continues on the development of a variety of test structures for use in providing feedback to the wafer fabrication activities at Stanford. The development of these test structures has paralleled the reestablishment of the wafer fabrication activities as well as the tighter process control requirements imposed by the development of a 2 μm CMOS.

4.1.1 Defect Density Test Structures

A set of test structures has been developed which allows the extraction of defect densities associated with step-coverage, contact opening, and shorts which may be encountered at various points in the fabrication sequence. Although meander patterns, contact strings, etc. provide a means of generating go/no-go statistics of these failures for a large coverage area, they provide little data as to where the defect actually occurs or even if a test failed because of one or many defects. The advantage of these types of tests, however, is the fact that they cover many sample sites and a large silicon area with a single measurement. The test structures that we are developing attempt to provide a greater degree of localization to better pinpoint the number and location of defects without totally sacrificing the ability to sample numbers of sample sites. These devices provide a means of addressing the test stucture, if desired, to help localize either the position or number of defects. The addressing circuitry allows a number of short test sections to be abutted together in a serial fashion to provide go/no-go testing over the full array. If the array fails, the short sections can be addressed either individually or in clusters using a binary search to progressively eliminate fault-free sections. The addressing circuitry is, of course, provided with a self-test mode to insure that we do not misinterpret peripheral failures as being due to the cell array. Initial designs have divided the array into only 16 sections to limit the number of pins required for address electronics so that we may maintain full compatability with the NBS 2 by 10 probe array.

4.1.2 2 Micron CMOS Test Structures

A full set of test devices has been developed to characterize the 2 μm CMOS process. These test structures attempt to study the performance of the n- and p-channel devices (both individually and taken together) as well as a number of the important parasitic effects which are increasingly important as the feature size decreases.

Even though CMOS is largely a ratioless technology (although it will not perform optimally with grossly mismatched Z/L ratios), we wish to examine inverter performance to pick the optimum size for both the n-channel and the p-channel devices. At larger feature sizes (~ 4 μm and greater), the p-channel device is often fabricated with a larger Z/L ratio than the n-channel device to offset the fact the electron mobility is greater than hole mobility at comparable doping levels. At a two micron feature size, this difference is not as large because (a) the n-channel device is more severely dominated by velocity saturation effects than the p-channel device and (b) for equivalent *drawn* gate lengths, the p-channel device will usually have a shorter effective channel length because of increased lateral penetration of the source/drain regions.

Latch-up is the parasitic device phenomenon in CMOS which deserves the closest scrutiny and good test structures are essential to the characterization of the latch-resistance of any CMOS technology. Because the operating voltages in small geometry integrated circuits have not been reduced, hot carriers are much more likely to initiate latch-up in CMOS. For these reasons, our test structures include a variety of devices which will be used to study the latch-up behaviour and hot carrier effects in 2 μm CMOS technology

4.1.3 Measurement Hardware

Our parametric measurement capabilities have been extensively modified during recent months. A HP 4145 Semiconductor Measurement System and a HP 6942 matrix switch have been added to our process/device characterization system. A HP 9845B controls these instruments as well as a Rucker and Kolls 1032 wafer prober and allows independent specification of the wafer stepping pattern, the matrix connections to the probe card, and the actual test to be performed. A preliminary investigation indicates that this software will be readily compatible with a HP9826/36, should the need arise. We now have improved software which drives the R & K 1032, so that our probing speed has been roughly doubled in recent months.

Staff: T. Walker, L. Gerzberg, W. Yarbrough

Related Efforts: M. Buehler (JPL), L. Linholm (NBS), V. Tyree (MOSIS), D. Trotter (Miss. St.)

4.2 The ICTEST System

The ICTEST system is a unified system for functional simulation and testing. The test is written in ICTEST, a superset of C extended to include testing primitives, data formatting, and mechanisms for specifying parallelism and pipelining. The test may then be targeted to run against a simulator (ESIM or TSIM) or a tester (MINIMAL, MEDIUM, or TEK S-3260). The MEDIUM tester is the testing workhorse; the TEK tester is intended primarily for performance measurement and functional testing at speed.

ICTEST itself has remained relatively stable over this period. We continue to use it to test our designs, including the MIPS test chips. Support for the clocking discipline is now substantially debugged, although we need to rethink our approach to qualified clocks and decide how they might be supported on the TEK (if that is indeed possible).

We are reducing the MEDIUM tester to a chip set. It will connect to a standard DEC DMA interface, and we shall distribute it to the community when it becomes available. The chip set that we have designed comprises 2 chip types, and a 64 pin tester will require a total of 3 chips. The tester control and pin electronic chips have both been designed and submitted for fabrication. Additionally, we have designed test chips for some new circuits that we need, including pads capable of driving 30mA loads. We have received the pin electronics chip and the pad test chip and partially tested them. They both seem to work correctly.

Staff: D. Boyle, D. Marple, R. Mathews, J. Newkirk, I. Watson

Related Work: FIFI Project (CalTech)

References: [Mathews 82], [Watson 82]

4.3 Clocking Discipline

We have developed a 2-phase clocking notation and an associated clocking discipline. The objective is to provide appropriate formal concepts for thinking about clocking in 2-phase systems, and to delineate a circuit syntax guaranteeing consistent clocking. The clocking discipline can also be co-opted to guarantee other forms of correctness, *e.g.*, freedom from charge sharing. The auditing tool *clockck* checks circuits extracted by the ESIM extractor for conformance to the discipline.

We have finished gleaning information from the Winter '82 testing class. Of 8 chips fabricated and tested, 3 designs had fatal clocking errors, and 2 had errors that could hurt performance. One of the three fatal cases was of the most interesting sort: a design that passed extensive simulation completely, but failed the clocking check and did not, in fact, work when fabicated.

Staff: R. Mathews, J. Newkirk, D. Noice

Related Efforts: Glasser's work (MIT)

References: [Noice 82]

4.4 Practical Testing

We have tested 30 more copies of a 10,000-transistor serial memory based on a 3-transistor RAM cell. The memory was originally intended as a step toward a serial signal processing system, but has actually proved to be test of our testing system and our understanding of the technology.

Of the 30 new parts from run M1DV, 40% are defect-free. Sixteen percent show anomalously low (less than 100-microsecond) storage times of the sort we reported previously. The remainder have failures that may be explained in terms of fabrication defects, with the exception of 2 chips that have the mysterious property that while every bit in the memory plane seems to be functioning correctly, when we apply error correction to this perfect data, errors result!

As a result of our frustration with short storage times, we have designed and tested a canary circuit that monitors storage times directly. The storage-time oscillator is a 3-stage ring oscillator, one stage of which is a storage node that is charged and allowed to relax toward ground.

We have tested 2 chips so far, one with very short (1 microsecond) storage times. The storage oscillator successfully indicates that the short-storage time chip is defective and that its companion is acceptable. Using optical injection to vary storage times over a large range, we have found that the storage oscillator on each chip predicts the storage time very precisely. The design seems to be insensitive to power supply variations, but we will need to test more chips before we are completely confident.

As a simple experiment in performance measurement, we have designed and tested an instrumented family of 7 PLAs with different loading characteristics. We have measured the performance of each path through each chip and computed regression lines to fit the observed data with delays predicted by τ models. The observed fits are very good, with correlation coefficients around .8 and derived τ's ranging from .25 to .57 nanoseconds. However, the intercepts of the regression line are non-zero, indicating systematic measurement errors specific to each member of the family. We are currently trying to chase these errors to ground.

Staff: G. Eckert, R. Mathews, J. Newkirk, T. Saxe, L. Shwetz, I. Watson

References: [Saxe 82]

5 Theoretical Investigations

5.1 Connected Components Algorithms

Finding the connected components of a graph, given its adjacency matrix, is a problem that has received much attention recently, but the best way to implement the algorithm in VLSI is not known. Using the AT^2 measure, it is possible to solve the problem in $n^{2+\epsilon}$ for an n node graph if one uses the mesh-of-trees, a layout that requires a great deal of area. Lipton and Valdes considered layouts that used area proportional to the number of nodes, and came up with an $n^{3+\epsilon}$ algorithm that is, unfortunately, not when-oblivious; the time at which inputs are required depends on the data. A. Siegel has recently invented an algorithm that runs in the same time as the Lipton-Valdes algorithm, is when-oblivious, and uses only n pads. The area is on the order of $n^{3/2}$, so its figure of merit is $AT^2 = n^{3.5+\epsilon}$, which is better than any known n-pad algorithm. The material has not yet been written down by the author, but a sketch appears in [Ullman 82b].

Staff: A. Siegel, J. Ullman

5.2 Modular Model of Event-based Concurrent Systems

The formal model we have been developing has two major components: a structural algebra for describing module interconnection structures, and a behavioral semantics that defines the function computed by a network of modules. Most of our work in the last six months has concentrated on the behavioral semantics.

As described in a previous report, the behavioral semantics associates with each module and network of modules:

- a functional mapping between partially ordered events at input and output ports,
- a domain constraint, specifying that certain output events must precede certain input events, and
- a functional constraint, specifying that certain input events must precede certain output events.

The domain constraint is essentially a statement of the conditions under which the module can be expected to work correctly. For example, it might require that no new input events arrive until after all outputs for the current input values have been produced. If the domain constraint is violated, the behavior becomes unpredicatable. The functional constraint, on the other hand, contains information about when a module will produce new output events. Thus the domain constraint tells what the module requires of its environment, and the functional constraint tells what it guarantees.

Our recent work has been particularly concerned with the problems of module substitution and the semantics of non-deterministic systems. The module substitution issue arises because we often wish to substitute one module for another in a network and need to know when this can be done without affecting the properties of the network. A simple criterion for such substitution is *semantic equivalence*. If two modules have the same functional mapping, domain constraint, and functional constraint, then one may replace the other without any change in the network's behavior.

In some cases, however, we need a more flexible criterion. We would like to be able to make a substitution so long as it allows the network to continue working correctly and produce the same output. This may be possible even with modules that are not identical. For example, suppose we have a system containing a module that can perform correctly as long as it is asked to buffer no more than three input elements at a time. (This would be expressed in the domain constraint.) If we replace this module with one that is identical except for the fact that it can buffer more items, then the new network should continue to work correctly. In general, we can always replace a module by one with a *weaker* domain constraint. If the environment of the original module guaranteed that the stronger domain constraint was satisfied, then the weaker one will necessarily be satisfied, and the composed system will continue to perform correctly. Likewise, we can always substitute a module with a *stronger* functional constraint, because if the original module operated in a way that satisfied the domain constraints of other parts of the network, then the (more constrained) new module must do so too. Thus we can perform such a substitution if the new module has an identical functional mapping, weaker (or identical) domain constraints, and stronger (or identical) functional constraints. The new network may not be equivalent to the old, but it will operate correctly in any environment where the old one does. This sort of substitution arises very naturally when the original system is viewed as a *specification* and the substitution represents an *implementation* of the specification.

The second problem we have considered is extending the semantics to non-deterministic systems. Non-determinism is a property of many concurrent systems. It may arise even in networks where all the primitive modules are deterministic; this is because the relative timing of events at different modules is unpredictable, and different timings may cause the system to produce different outputs. A non-deterministic module can be described by altering the functional mapping to give, for each input, a *set* of possible outputs. There are several technical problems that must be resolved in this sort of definition. The most significant is being able to guarantee that loop-feedback (the operation that sends some of a network's output to its own input ports) is always well-defined. By modifying the approach of Plotkin and Smyth, which deals with non-determinism in state-oriented rather than event-oriented models, we have been able to solve these problems and develop a mathematically sound semantics for non-deterministic networks.

Staff: S. Owicki and N. Yamanouchi

5.3 Defect Tolerance in Array Architectures

We have developed a new body of theory treating the effect that defects have on yield of array architectures. The theory addresses such issues as whether it is possible to find chains or arrays of working elements embedded in a large array and what reconfiguration capabilities must be available for the yield of the reconfiguration process to be non-zero.

For the problem of finding a connected chain of working elements in a square array, we have developed a new algorithm that requires time linear in the number of elements to be chained. We have also made progress on the problem of finding an array of working elements embedded within a larger array by tightening the bounds determining when such reconfiguration is possible. We are beginning some new work investigating the effects of defects in the interconnect itself.

Staff: A. El Gamal, J. Greene

References: [GreeneEl Gamal 82]

5.4 Wiring Area for Gate Arrays

By applying statistical modeling techniques, we have developed a body of theory that predicts how to realize a given function in a gate array with smallest overall die size. The central result is that it is preferable to have a smaller gate array with a larger number of tracks in between blocks, thereby permitting higher overall use of the array elements, rather than sparsely using a larger array. These theoretical results are borne out by a large body of empirical data collected by IBM.

Staff: A. El Gamal

References: [EL Gamal 82]

6 The SUN Workstation: File System Development

The SUN represents a radical departure from the customary workstation design in that it does not have a local disk. A typical SUN workstation at Stanford has 256K bytes of memory, a frame buffer with some additional memory to hold the raster image, and a high-performance Ethernet link. In order to use the SUN as a workstation for VLSI design or other applications, it must be possible to read and write file storage from software resident in the SUN.

Our initial approach, which permits us to run simple software on the SUN, was to implement a page-at-a-time file server called a Leaf Server, which ran on a supporting computer (usually a Vax) and provided disk page access in response to request packets. This server and its development were reported in last year's progress report, and we have done little work to it since. It is worth mentioning that the Unix-based Leaf Server that we wrote last year was made available to other Arpa-supported users of Xerox 1100 Lisp Workstations, and (after suitable modifications at ISI to make it compatible with the Xerox equipment) it provides a valuable disk support facility to AI programmers using the 1100's.

Our experience with the Leaf Server approach to Remote File Access demonstrated conclusively that a single-host file server was not an adequate level of file support for a network machine participating in a distributed system. We experimented for a few months with changes that could be made to the Unix file system or to the behavior of the Unix Leaf Servers that would make a more reasonable distributed file system available to SUN users. We abandoned this approach for three reasons:

- The Unix file system does not map neatly to a distributed environment. At the design level, it assumes that there is at most one copy of any file, and that the entire file system is tree-structured. It is difficult to modify the Unix kernel to think that parts of its file systems are on other machines, though at the 1981 SIGOPS conference some Bell Labs researchers reported having accomplished it (with a severe degradation in performance.) At the implementation level, its locking mechanisms are unreliable and the fixed I/O table sizes in the kernel provide unreasonable fixed bounds on the total number of files that can be accessed simultaneously on a given server host. We thought that we could get by with a combination of a few kludged Unix file systems for the first generation of SUN software, but the problems overwhelmed us.

- Leaf-Server access to files on a time-shared Unix system was a second class citizen. We frequently found the need to log a job on to the host Unix, probe around the file system, then log off and resume a stopped SUN job that was having file problems. This problem could be bandaged by providing a set of file utility programs resident on the SUN, but the essence of the problem is that a Unix file system is not very suitable for a non-Unix-like operating system; we do not intend to run Unix on our SUNs.

- Even with just 4 server hosts available (3 Unix and one Tenex), the amount of context that needed to be maintained in a user's head was overwhelming. The lack of automatic location, migration, or replication facilities made it particularly difficult to find files whose precise location was not known.

We therefore, reluctantly, concluded that we were going to have to design and implement our own file system to our own specifications. This file system would be network-wide, provide a uniform set of access mechanisms and management tools, and be implemented on a variety of file computers.

We have settled on a multi-level design based around a central archival file system and distributed cached copies. We are implementing the design "from the inside out", starting with the reliable archival part and working towards the fast cache servers; the reasoning behind that decision being that it is better to have a slow reliable file system than a fast unreliable file system during the development phase. We intend to present a paper on this system at the 1983 SIGOPS conference, whose paper submittal deadline is in January, and we intend to have the archival server working and the migration protocols designed by that time. The initial implementation is taking place on our time-shared VAX, but we hope to move to a dedicated machine with a larger disk as soon as it becomes available.

Staff: J. Mogul, B. Reid

References: [Baskett 82]

6.1 Computer support for a Fast Turnaround Laboratory

We have completed the planning and exploratory stages of a project to provide extensive automation and computer support for the Fast Turnaround Laboratory. This ambitious interdisciplinary project (involving researchers from Computer Systems, Integrated Circuits, and Solid State laboratories) will provide control, documentation, training, portability, repeatability, and efficiency in the area of IC fabrication processes.

As a result of this exploration, we have isolated the following goals for this project:

- Automatic control of the IC fabrication processing equipment.
- Integration of fabrication control with simulation control, to run simulation and fabrication in parallel.
- A transportable, repeatable means for recording a fabrication process.
- The ability to repeat an arbitrary process on demand, for demand production of parts.
- The ability to manage and schedule a single IC fabrication line for multiple purposes.

- Computer aids for training and documentation.
- General data processing and database support for analytic work in the Laboratories.

We recognize that particular technical achievements will contribute to the realization of several of these goals. The most important of these is the development of a language for the representation of fabrication processes. When this language exists, it can be used as input to the control system, as input to the simulation system, as the contents of an archive for demand production of parts, and as a basis for training and documentation aids. Furthermore, the nature of this language will color most of the other work.

We have therefore spent several months on the preliminary design of such a language, our Language for Specifying Manufacturing Processes, or LSMP. (We intend to change its name before its specification is published, but we are for the moment calling it LSMP internally). This language resembles Ada semantically, but contains built-in type support for non-numeric types pertaining to manufacturing, and contains extension and package mechanisms suitable for the description of typed objects whose physical existence is outside the controlling computer, e.g. furnaces. We quickly found that two languages were necessary, one to describe the manufacturing process and another to describe the fabrication line itself. This second language corresponds very much to the microcode found on conventional computers, and it is used to implement a somewhat abstract instruction set, into which the LSMP is compiled. A compiled LSMP program can operate an automated fabrication line with the appropriate microcode, or it can operate a simulation system (one component of which would be programs like SUPREM) with a different set of microcode. Other microcode would be written to permit experimentation and testing of new processes before actually turning them loose on the fabrication equipment.

Only by implementing all pieces of this system and actually using it to manufacture parts can we satisfy ourselves that it is complete; we would therefore want to do an automating implementation even if that were not one of the goals of the project.

Having completed the first level of implementation, we intend to write software that amounts to a distributed time-shared operating system for the fabrication line; this will permit multiple independent fabrication processes, or laboratory experimentations, to be run on the same fabrication line simultaneously just as a time-shared computer is now capable of running independent programs simultaneously. This operating system will also take responsibility for the long-term scheduling and priority realization.

We consider that a system like this will be a superb testbed for explorations in knowledge-based systems for training and diagnosis, and for applications of interactive graphics, computer aided instruction, and reliable models of computation. We therefore hope to attract talented graduate students from various areas related to computer science, in addition to the core of people knowledgeable in IC fabrication, to this project. Currently Prof. Brian Reid is spending 90% of his research time on this project, and one graduate student, Harold Ossher, is working on it full time. Several more graduate students are eager to join as soon as funding becomes available to them.

7 Fast Turn-Around Laboratory

Activities if the Fast Turn-Around Laboratory have concentrated on the characterization of a significant amount of fabrication, mask making, and testing equipment followed by the subsequent incorporation of these items into the primary NMOS and CMOS processes. The following subsections of this report will detail the performance of many of these pieces of equipment and indicate how they have enabled us to increase the quality of our NMOS/CMOS wafer production and the quality of our device research.

7.1 Wafer Fabrication

During this period upgrading the processing lab has continued, in order to meet the project goals of establishing standard 2 micron CMOS and NMOS processes. Much of this effort was directed at bringing online and characterizing the equipment ordered during the previous reporting period. This equipment includes three low pressure *chemical vapor deposition* (LPCVD) systems, two plasma dry etchers, and a photoresist processing system. In addition, a 1:1 projection alignment system capable of 1.25 micron linewidths was installed and is being characterized. Finally a sputtering system and linewidth measurement system have been ordered.

7.1.1 Low-Pressure Chemical Vapor Deposition

The older multi-purpose atmospheric Epi/CVD system has been replaced by three (poly, nitride and oxide) dedicated LPCVD sytems. Low pressure deposition offers significant improvements over atmospheric deposition as following:

1. Improved thickness uniformity (2% vs. 15%) due to the 10^3 improvement in the gaseous diffusion coefficients.

2. Fewer particle generated defects due to vertical wafer positioning and the "hot" wall deposition on the tube in the LPCVD system, versus horizontal positioning and "cold" wall deposition in the atmospheric system.

Nitride CVD is essential to the local oxidation isolation process used for NMOS and CMOS. Use of the LPCVD nitride system is now standard in our lab, and is capable of depositing the required 800 Angstrom film to a uniformity of 2%. The typical deposition rate is 36 Angstrom/min. The gas flows are 20 SCCM of dichlorosilane and of 60 SCCM of ammonia. The furnace temperature is 790 degrees Celsius and the deposition pressure is 500 mT.

Doped polysilicon is used as the gate material and as a conductor in both NMOS and CMOS. The change over to LPCVD not only improved thickness uniformity and particle control, but also reduces the grain size because of the lower deposition temperature (620 degrees C for LPCVD vs. 900 degrees C for APCVD). The smaller grain size is important since it reduces grain boundary related oxide defects, and reduces poly edge roughness. The poly deposition is also done at a pressure of 500 mT and with a silane flow of 30 SCCM. The deposition rate is 90 Angstrom/min.

The last of the new LPCVD systems which is operational is the low temperature oxide (LTO) system. This phosphorus-doped oxide deposited at 450?C is intended to replace the atmosphere-deposited vapox layer which is reflowed to improve step coverage. Initial undoped LTO films show excellent uniformly (roughly 1%) and conformal step coverage. The deposition pressure is 400 mT and the flows are 60 SCCM of silane and 95 SCCM of oxygen. Phosphorus doping during deposition is now being characterized.

7.1.2 Plasma Etching

In order to achieve line widths of three microns and less, wet etching must be replaced with dry etching techniques. For poly and nitride etching, a Drytek RIE 100 etcher has been purchased. This etcher,which is in routine operation, operates in the plasma etch mode. It is fully automated to eliminate handling-induced defects and to give better process control. A key feature is its use of a interferametric laser end point detection system. For poly etching, a controlled slope process has been achieved which gives a 70 degree wall slope with a critical dimension loss of only 0.1 micron per edge. Controlled slope, as opposed to pure 90 anisotropic etching, is desirable for step coverage needs. The present process uses a mixture of C_2ClF_5 and SF_6 both at flows of 50 SCCM. The pressure is controlled at 150 mT and the RF power density is at 0.3 watts per cm^2. For this process, the etch rate is 2000 Angstrom/min, and the selectivity of poly to both the resist and the underlying oxide is 20:1.

For the less critical etching of silicon nitride films, an isotropic process is used. This process uses a mixture of CF_4 and O_2 with flows of 90 and 10 SCCM, repsectively. The pressure is again set at 150 mT and the RF power is again 0.3 watts per cm^2. The etch rate is 160 Angstrom/min with selectivity to oxide of 4 to 1.

For the etching of SiO_2, Branson/IPC has given us one of their Sigma 80 etchers. This unit is an automated single-wafer-at-a-time machine. The SiO_2 process uses a mixture of C_2F_6, CHF_3 and He at respective flows of 300, 300, and 3000 SCCM. The pressure used is 10 Torr and the power density is 5 watt/ cm_2. The etch rate on thermal oxide is 5000 Angstrom/min and the selective to both resist and silicon is roughly 7:1. The resulting oxide wall slope is currently 75 degrees. The process is being modified to improve selectivity, uniformity, and wall slope.

7.1.3 Photolithography

An automated photoresist resist processor has been brought on line. This machine offers full cassette operation with microsprocessor control of all functions and will significantly reduce resist related defects. Functions include priming, resist coating, "puddle" development, and microwave baking. The processor is in standard use except for the microwave bake feature, which will replace the use of the resist bake oven when characterized.

An Ultratech 1:1 projection stepper has recently been installed and is being characterized. This machine offers significant area utilization improvements over our Canon 4:1 manual stepper, which is limited to a *total* exposed area of only 3 cm by 3 cm. The Ultratech is a fully automated state-of-the-art optical alignment system which has auto focus, alignment, and load. It has a working resolution of 1.25 microns, with an alignment accuracy of 0.14 microns. The pellicle protected reticle has four selectable fields which offer a maximum *unique* silicon area of 6 cm^2. An advantage of this stepper over the popular 10:1 systems is that it uses two wavelengths (405 and 436 nm) and thus is less susceptable to standing wave problems.

An optical linewidth measurement system has been ordered. This auto focussing unit will be used to obtain tighter control of linewidths during processing.

These pieces of fabrication equipment have been used in the fabrication of both NMOS and CMOS device wafers. The performance of the CMOS devices will be discussed in the Device Research subsection.

7.2 Electron Beam Lithography

The principal activity of the E-beam lithography group involved the installation, characterization, and, finally, the on-site acceptance of the Perkin-Elmer/ETEC MEBES machine. The on-site acceptance tests included extensive testing of the registration (alignment) accuracy of the MEBES machine in anticipation of its use as a direct-write lithography tool. During the period when the E-beam machine was undergoing acceptance tests, we were characterizing the resist developing and chrome etching systems in our lab. The first use of the ability to develop and etch chrome plates was to produce the 2 μm CMOS test plates for Jim Pfiester. These plates were actually written at P-E/ETEC, but developed and etched at Stanford, before our machine had passed on-site acceptance.

At present, we are in the process of making masks for the Ultratech 900 projection lithography system. The

Ultratech is a 1:1 wafer stepper whose mask requirements are somewhat different from that of a Perkin-Elmer 140/240, so we are at present carefully formatting these plates manually rather than using the MOSIS service. It does appear, however, that we will be able to automatically place the desired dice, alignment marks, and scribe lanes. This first mask set will include Jim Clark's Geometry Engine, several of John Hennessy's MIPS test vehicles, and the two chips designed by Newkirk/Mathews/Watson that comprise a medium tester.

In addition to working with the MEBES machine, we have been developing a tri-level resist technology for use in direct-write applications. The principal need for a multi-layer direct-write technology is to provide a thick, chemically resistant "working" layer of resist on the wafer and yet maintain a thin, high resolution layer of resist for the actual electron patterning.

In our tri-level, the underlying layer of resist is 1.2 μm of AZ-1470 which has been baked at 200 degrees Celcius to remove any photosensitivity. The AZ-1470 has very good chemical resistance, planarizes the wafer surface, and reduces secondary electron backscatter (compared to a silicon substrate) which can be a source of image degradation. The second layer is a thin (500-800 Angstrom) layer of poly-Si which will ultimately be used as the intermediate masking layer in the process of transferring the pattern from the PBS electron resist to the underlying AZ 1470. We wish to keep this layer thin and of low atomic mass to minimize secondary electron image degradation. We have used both evaporation and plasma-enhanced chemical vapor deposition to deposit this layer. Researchers who are exploring tri-level resist for use in optical lithography favor the use of SiO_2 as the intermediate material because of its low index of refraction. For electron exposure, however, the slight conductivity of poly-Si is preferable to reduce charging effects. The top layer is 0.4 μm of PBS electron resist which is coated, exposed, and developed as if on a chrome blank. This pattern is then transferred to the poly-Si using plasma etching (an anisotropic CF_4 + 4% O_2 etch is adequate because the poly-Si is so thin) which in turn serves as a mask for O_2 reactive ion etching. Using a partial pressure of 6 μm Hg of O_2, we have achieved nearly vertical sidewalls in features 0.5 μm wide separated by 0.5 μm.

We have been investigating two reactive ion etching systems for routine use in this application: one is manufacturered by Materials Research Corporation and the other is manufacturered by Drytek.

7.3 Device Research

Many different aspects of our device research program have an impact on the development of a 2μm CMOS technology. Jim Pfiester has designed a mask set containing a wide variety of CMOS test structures to aid him in the development of this process. The mask set was generated without the use of bloats or shrinks during mask making to provide an accurate measure of what bloats/shrinks will be required to produce the "drawn" dimension in silicon. This is an n-well process which has a surface concentration that does not require a channel stop for the p-channel devices. The n-channel drain/source junction depths are only 0.3 μm deep and the p-channel drain/source region are about 0.55 μm deep. The measured electrical channel length for the n-channel and p-channel devices was 1.6 and 1.1 μm for a *drawn* dimension of 2.0 μm, indicating that our poly-Si plasma etching is providing an anisotropic etch profile because most of the difference between the effective and drawn channel lenghts is due to the lateral diffuaion from the drain and source regions.

Electrically, the n-channel devices look very good in terms of leakage current, threshold voltage. Previous runs had indicated that drain-induced barrier lowering (DIBL) is the most stringent test of these devices. These devices exhibit very good drain-induced barrier lowering properties. The threshold voltage of the p-channel devices are as we expect them to be. Unfortunately, these first devices had a parasitic leakage current from drain to substrate along the surface of the n-well which is superimposed on the actual drain/source current. The cause of this leakage is under close scrutiny and two sets of additional CMOS wafers are nearing the end of the fabrication sequence.

In order to help in the investigation of latch-up in short channel CMOS structures, we have initiated a program to investigate the properties of devices as a function of temperature. From a device physics standpoint, it is extremely desirable to have such a capability in order to establish the activation energy of the phenomena under investigation. We are at present constructing a test fixture which will allow us to scan the temperature of a packaged device from 77 degess Kelvin to 100 degrees Celsius.

Staff: J. Shott, J. McVittie, E. Wood, K Saraswat, R. Castellano, F. Pease, D. Dameron, C.-C. Fu, P. Jerabek, J. Plummer, J. Pfiester, T. Nguyen, L. Lewyn, J. Marshall, A. Henning, D. Gardner

8 Other Projects

8.1 Polygon Package and Design-Rule Checker

For some time now, we have made available a high-quality design-rule checker based on our polygon package. It derives circuit connectivity information to prevent reporting of false separation errors between electrically connected components. This checker is used by our design classes and for our research and has been heavily tested by 80+ designers. We have recently added support for buried contacts in nMOS.

We have developed and tested an analogous checker for the JPL bulk CMOS rules. It has checked the designs submitted by Stanford, MIT, and Lincoln Labs for the bulk CMOS run. We have distributed the CMOS checker to JPL and MIT.

Staff: D. Noice

8.2 Cell Library

The nMOS cell library is now being prepared for publication by Addison-Wesley as a companion to the Mead and Conway text. Accordingly, we are cleaning up the documentation, correcting minor design-rule violations, and thoroughly checking the cells. Some new cells will be included, such as LSSD PLA buffers.

We have designed and submitted bulk CMOS cells to form the basis of a CMOS cell library. The new cells are pads, PLA designs, and counters.

Staff: R. Mathews, J. Newkirk, J. Shott, T. Walker

8.3 Modifications to MIT Circuit Extractor

We have integrated the MIT circuit extractor with our CLL/CIF processing software, resulting in an order-of-magnitude improvement in extraction speed. Previously, the 10,000-transistor serial memory required several hours to extract; extraction now requires approximately 10 minutes. We will distribute the extractor if there is sufficient interest; however, prospective users should be aware that our CIF processing system is restricted to Manhattan-only, rectangle-only designs.

Staff: J. Newkirk, T. Saxe, S. Taylor

References

[Baskett 82] Bechtolsheim, A., Baskett, F., and Pratt, V.
The SUN Hardware Architecture.
Technical Report 229, Computer Systems Laboratory, Stanford University, March, 1982.

[Bobrow 81] Bobrow, D. and Stefik, M.
The LOOPS Manual.
Technical Report KB-VLSI-81-13, Knowledge-Based VLSI Design Group Memo, 1981.

[Brown 82] Brown, H. and Stefik, M.
Palladio: An Expert Assistant for Integrated Circuit Design.
Technical Report HPP-82-5, Stanford University Heuristic Programming Project, 1982.

[Bryant 81] Bryant, R.
A Switch-Level Model of MOS Logic Circuits.
In *Proceedings of the First International Conference on VLSI.* Academic Press, 1981.

[Clark 82] Clark, J.
The Geometry Engine: A VLSI Geometry System for Graphics.
In *Proc. SIGGRAPH '82.* ACM, 1982.

[Davis, T. and Clark, J. 82]
Davis, T. and Clark, J.
YALE User's Guide: A SILT-Based VLSI Layout Editor.
Technical Report 223, Computer Systems Laboratory, Stanford University, October, 1982.

[EL Gamal 82] El Gamal, A.
Notes on Gate Array Wiring Area Prediction.
1982.
ISL VLSI File #102282.

[Floyd 82] Floyd, R. W. and Ullman, J. D.
The Compilation of Regular Expressions into Integrated Circuits.
JACM 29(3):603-622, July, 1982.

[Genesereth 80] Genesereth, M., Griner, R. and Smith, D.
MRS Manual.
Technical Report HPP-80-24, Stanford University Heuristic Programming Project, 1980.

[Gerring 81] Gerring, P.
HILGA: A high level, representation independent, object-oriented graphics package for Interlisp-D.
Technical Report KB-VLSI-81-12, Knowledge-Based VLSI Design Group Memo, 1981.

[GreeneEl Gamal 82]
Greene, J. and El Gamal, A.
Area and Time Penalties for Restructurable VLSI Arrays.
1982.
submitted to the Third Caltech Conference on VLSI, March 1983.

[Gross 82] Gross, T.R. and Hennessy, J.L.
Optmizing Delayed Branches.
In *Proceedings of Micro-15.* IEEE, October, 1982.

[Hemachandra 82]
Hemachandra, L.
GRY: a PLA Minimizer.
1982.
Unpublished memorandum, Stanford Univ., Dept. of C. S.

[Hennessy 83] Hennessy, J.L. and Gross, T.R.
Postpass Optimization of Code in the Presence of Pipeline Constraints.
ACM Trans. Prog. Lang. and Sys., 1983.
accepted for publication.

[HennessyJouppiPrzybylski 82]
Hennessy, J.,Jouppi, N., Przybylski, S., Rowen, C., Gross, T., Baskett, F., and Gill, J.
MIPS: A Microprocessor Architecture.
In *Proceedings of Micro-15*, pages 17-22. IEEE, October, 1982.

[Karplus 82] Karplus, K. J.
CHISEL: an Extension to the Programming Language C for VLSI Layout.
PhD thesis, Stanford Univ., Sept., 1982.

[Novak 82] Novak, G.
GLISP: A High-Level Language for A. I. Programming.
In *Proceedings of the National Conference on Artificial Intelligence.* AAAI, 1982.

[Smith 82] Smith, L., Saxe, T., Newkirk, J., Mathews, R.
A New Area Router, the LRS Algorithm.
In *Proc. ICCC*, pages 256-259. New York, September, 1982.

[Stefik 82a] Stefik, M., Bobrow, D., Bell, A., Brown, H., Conway, L. and Tong, C.
The partitioning of concerns in digital system design.
In Penfield, P. (editor), *Proceedings, Conference on Advanced Research in VLSI.* Artech House, 1982.

[Stefik 82b] Stefik, M., Bell, A. and Bobrow, D.
Rule-Oriented Programming in LOOPS.
Technical Report KB-VLSI-82-19, Knowledge-Based VLSI Design Group Memo, 1982.

[Tong 82] Tong, C.
A Framework for Design.
Technical Report HPP-82-11, Stanford University Heuristic Programming Project, 1982.

[Trickey 82] Trickey, H. W.
Good Layouts for Pattern Recognizers.
IEEE Trans. on Computers C-31(6):514-520, June, 1982.

[TrickeyUllman 82]
Trickey, H. W. and Ullman, J. D.
A Regular Expression Compiler.
In *Proc. COMPCON 82.* IEEE, Feb., 1982.

[Ullman 82a] Ullman, J. D.
Combining State Machines and Regular Expressions for Automatic Synthesis of VLSI Circuits.
Technical Report STAN-CS-82-927, Dept. of C. S., Stanford Univ., Sept., 1982.

[Ullman 82b] Ullman, J. D.
Algorithms and Complexity Issues in VLSI.
1982.
Department of Computer Science, Stanford Univ.

[Wolf 82] Wolf, W., Newkirk, J., Mathews, R., Dutton, R.
Dumbo, a Schematics-to-Layout Compiler.
1982.
submitted to the Third CalTech Conference on VLSI, Mar. 1983.

Publications

[Brown 82] Brown, H. and Stefik, M.
Palladio: An Expert Assistant for Integrated Circuit Design.
Technical Report HPP-82-5, Stanford University Heuristic Programming Project, 1982.

[Davis, T. and Clark, J. 82]
Davis, T. and Clark, J.
YALE User's Guide: A SILT-Based VLSI Layout Editor.
Technical Report 223, Computer Systems Laboratory, Stanford University, October, 1982.

[DavisClark 82] Davis, T. and Clark, J.
SILT: A VLSI Design Language.
Technical Report 226, Computer Systems Laboratory, Stanford University, October, 1982.

[EL Gamal 82] El Gamal, A.
Notes on Gate Array Wiring Area Prediction.
1982.
ISL VLSI File #102282.

[Floyd 82] Floyd, R. W. and Ullman, J. D.
The Compilation of Regular Expressions into Integrated Circuits.
JACM 29(3):603-622, July, 1982.

[GreeneEl Gamal 82]
Greene, J. and El Gamal, A.
Area and Time Penalties for Restructurable VLSI Arrays.
1982.
submitted to the Third Caltech Conference on VLSI, March 1983.

[Gross 82] Gross, T.R. and Hennessy, J.L.
Optimizing Delayed Branches.
In *Proceedings of Micro-15.* IEEE, October, 1982.

[Hemachandra 82]
Hemachandra, L.
GRY: a PLA Minimizer.
1982.
Unpublished memorandum, Stanford Univ., Dept. of C. S.

[Hennessy 83] Hennessy, J.L. and Gross, T.R.
Postpass Optimization of Code in the Presence of Pipeline Constraints.
ACM Trans. Prog. Lang. and Sys., 1983.
accepted for publication.

[HennessyJouppiPrzybylski 82]
Hennessy, J.,Jouppi, N., Przybylski, S., Rowen, C., Gross, T., Baskett, F., and Gill, J.
MIPS: A Microprocessor Architecture.
In *Proceedings of Micro-15*, pages 17-22. IEEE, October, 1982.

[Karplus 82] Karplus, K. J.
CHISEL: an Extension to the Programming Language C for VLSI Layout.
PhD thesis, Stanford Univ., Sept., 1982.

[MathewsWatson 82]
Mathews, R. Watson, I. and Wolf, W.
Testing Chips using ICTEST Version 2.5.
1982.
ISL VLSI File # 012782.

[Noice 82] Noice, D., Mathews, R., Newkirk, J.
A Clocking Discipline for Two-Phased Digital Systems.
In *Proc of ICCC*, pages 108-111. New York, September, 1982.

[Saxe 82] Saxe, T.
Testing the High Yield Memory.
1982.
ISL VLSI File #821019.

[Smith 82] Smith, L., Saxe, T., Newkirk, J., Mathews, R.
A New Area Router, the LRS Algorithm.
In *Proc. ICCC*, pages 256-259. New York, September, 1982.

[Tong 82] Tong, C.
A Framework for Design.
Technical Report HPP-82-11, Stanford University Heuristic Programming Project, 1982.

[Trickey 82] Trickey, H. W.
Good Layouts for Pattern Recognizers.
IEEE Trans. on Computers C-31(6):514-520, June, 1982.

[TrickeyUllman 82]
Trickey, H. W. and Ullman, J. D.
A Regular Expression Compiler.
In *Proc. COMPCON 82.* IEEE, Feb., 1982.

[Ullman 82a] Ullman, J. D.
Combining State Machines and Regular Expressions for Automatic Synthesis of VLSI Circuits.
Technical Report STAN-CS-82-927, Dept. of C. S., Stanford Univ., Sept., 1982.

[Ullman 82b] Ullman, J. D.
Algorithms and Complexity Issues in VLSI.
1982.
Department of Computer Science, Stanford Univ.

[Watson 82] Watson, I., Newkirk, J., Mathews, R. and Boyle, D.
ICTEST: A Unified System for Functional Testing and Simulation of Digital ICs.
In *Proc. Cherry Hill Int'l Test Conference.* Philadelphia, 1982.

[Wolf 82] Wolf, W., Newkirk, J., Mathews, R., Dutton, R.
Dumbo, a Schematics-to-Layout Compiler.
1982.
submitted to the Third CalTech Conference on VLSI, Mar. 1983.